ANÁLISIS DEL MOVIMIENTO EN SECUENCIAS DE IMÁGENES

María Victoria González García

DOBLE GRADO EN INGENIERÍA INFORMÁTICA Y MATEMÁTICAS
FACULTAD DE INFORMÁTICA
UNIVERSIDAD COMPLUTENSE DE MADRID

TRABAJO DE FIN DE GRADO
DEPARTAMENTO DE INGENIERÍA DE SOFTWARE E INTELIGENCIA ARTIFICIAL

Junio 2016

Director:
Gonzalo Pajares Martinsanz

Índice general

Agradecimientos

En primer lugar quiero agradecer a mis compañeros de clase y amigos, con los que he compartido muchos momentos, malos y buenos, durante estos años de carrera. Por todo lo que me han ayudado durante esta etapa, dentro y fuera de clase. Me gustaría agradecer también a Gonzalo Pajares su dedicación y tiempo en la elaboración de este trabajo. Y, por supuesto, a mi familia, en especial a mi madre, por su apoyo incondicional.

Resumen

Este trabajo presenta el desarrollo de una aplicación destinada al análisis de secuencias de imágenes para la detección de movimiento en la escena. Se trata de un campo importante de la Visión Artificial, con múltiples aplicaciones entre las que se encuentra la videovigilancia con fines de seguridad, el control de tráfico, el movimiento de personas o el seguimiento y localización de objetos entre otras muchas. Para ello se utilizan métodos de análisis como son el de Lucas-Kanade y Gauss-Seidel, que obtienen el denominado flujo óptico. Éste describe el movimiento que ha tenido lugar entre las imágenes y su fundamento estriba en la determinación de las variables espaciales y temporales en las imágenes, siendo precisamente la variable temporal la que introduce el concepto fundamental para el análisis del movimiento a partir de las imágenes captadas en diferentes instantes de tiempo dentro de la secuencia analizada.

Para el desarrollo de la aplicación se han utilizado técnicas propias del tratamiento de la Visión Artificial, así como la metodología proporcionada por la Ingeniería del Software. Así, se ha realizado una especificación de requisitos, se ha elaborado y seguido un plan de proyecto y se ha realizado un análisis de alto nivel, que se materializa en el correspondiente diseño e implementación, junto con las pruebas de verificación y validación, obviamente adaptados en todos los casos a las dimensiones del proyecto, pero que establecen claramente los planteamientos básicos para el desarrollo de una aplicación a nivel empresarial.

La aplicación planteada se enmarca perfectamente dentro del paradigma, hoy en día en pleno auge, conocido como el Internet de las Cosas (IoT). El IoT permite la intercomunicación entre dispositivos remotos, de forma que mediante la correspondiente comunicación a través de conexiones a Internet es posible obtener datos remotos para su posterior análisis, bien en nodos locales o en la nube, como concepto íntimamente relacionado con el IoT. Éste es el caso de la aplicación que se presenta, de suerte que los métodos de procesamiento de las imágenes pueden aplicarse localmente o bien transmitir las mismas para su procesamiento en nodos remotos.

Palabras Clave

Análisis del movimiento en imágenes, vídeo-secuencias, flujo óptico, visión artificial, ingeniería del software, internet de las cosas.

Abstract

This work presents the development of an application to analyze image sequences in order to detect movement in the scene. This is an important field in Computer Vision, with multiple applications such as video surveillance for security purposes, control of traffic and movement of people or tracking and localization of objects, among others. In order to do that, the Lucas-Kanade and Gauss-Seidel methods obtain the so called optic flow. It describes the movement that has taken place between the images, and its basis lies in the determination of the spatial and temporal variables in the images, being precisely the temporal variable the one that introduces the essential concept for motion analysis from the images captured at different times within the analyzed sequence.

Typical techniques of Computer Vision have been used for the development of the application, as well as the methodology provided by the Software Engineering. Therefore a software requirements specification has been made, a project management plan has been developed and followed, and a high level design has been produced, which is materialized in the corresponding design and implementation, along with verification and validation tests. They are obviously adapted in all cases to the dimensions of the project, clearly establishing however the basic approaches to the development of an application on the business level.

The considered application falls perfectly under the paradigm which is nowadays at its peak known as the Internet of Things (IoT). The IoT allows the intercommunication between remote devices, so that by means of the corresponding communications through the Internet it is possible to gather remote data

for a later analysis, either in local nodes or in the cloud, as a concept closely related to the IoT. This is the case of the present application, so that the methods of image processing can be applied locally or rather broadcast the images to be processed on remote nodes.

Key Words

Motion analysis in images, video-sequences, optical flow, computer vision, software engineering, internet of things.

Lista de acrónimos

A continuación se proporciona un listado de definiciones, términos y acrónimos que aparecen en el proyecto.

SRS Especificación de requisitos de software.
PF Puntos de Función.
VO Valor Optimista.
VPR Valor más probable.
VPE Valor Pesimista.
VE Valor Esperado.
F_i Factores de Ajuste de Valor.
E Esfuerzo.
Pr Productividad organizacional promedio.
CPTP Coste Presupuestado del Trabajo Planificado.
CPTR Coste Presupuestado del Trabajo Realizado.
VP Variación de la planificación.
IRP Índice de Rendimiento de la Planificación.

Capítulo 1

Perspectivas del proyecto

1.1. Introducción

El propósito del proyecto es el desarrollo de una aplicación informática destinada al análisis de secuencias de imágenes, con el fin de detectar el movimiento producido por objetos en la escena.

1.2. Motivación, objetivos y propuestas

El objetivo de la aplicación desarrollada es la detección de movimiento a través del análisis del flujo óptico obtenido a partir de los algoritmos de Lucas-Kanade y Gauss-Seidel, obteniendo tanto el sentido como la magnitud del movimiento.

La principal utilidad de la aplicación es la videovigilancia, tomando como entrada secuencias de vídeo a partir de la toma de imágenes mediante cámaras de seguridad. En particular su uso se orienta a la detección de intrusos en recintos controlados. Otras aplicaciones serán el control del tráfico (su flujo y congestión) o bien el seguimiento de objetos de interés.

La aplicación, que se desarrollará en lenguaje Matlab [1], posee un interfaz para interacción hombre-máquina de forma que permite la selección manual de uno de los dos algoritmos mencionados para el estudio de un par de imágenes o

bien la secuencia de vídeo completa, a la vez que solicita los parámetros necesarios y devuelve el resultado para su visualización. El diseño software se ha realizado con una gran flexibilidad con el fin de que en el futuro pueda integrarse en una aplicación más amplia de vigilancia que analice vídeos en tiempo real y de forma automática.

1.3. Organización de la memoria

El siguiente capítulo describe el diseño de la aplicación. Contiene la especificación de los requisitos, el plan del proyecto y el diseño de alto nivel. Para la especificación de requisitos y el plan se han seguido, adaptándolos al presente proyecto, los IEEE 830 y 1058 respectivamente ([5],[6]).

El capítulo 3 describe los métodos de análisis del movimiento utilizados por la aplicación.

El cuarto capítulo recoge los detalles de la implementación y los resultados obtenidos.

En el último capítulo se propone trabajo futuro y se presentan las conclusiones.

Capítulo 2

Planteamiento, análisis y diseño

2.1. Requisitos

A continuación se presenta la Especificación de Requisitos, que detalla los servicios y restricciones del sistema software. En particular, identifica la funcionalidad, las interfaces externas, las prestaciones, los atributos y las restricciones de diseño.

2.1.1. Introducción

Esta sección proporciona una visión general de la Especificación de Requisitos, detallando el propósito y ámbito del documento, y de la aplicación en general.

Propósito y visión general

La Especificación de Requisitos recoge la especificación de la aplicación que se pretende desarrollar, constituyendo la fase previa al diseño e implementación de la aplicación. Se busca dar al cliente una visión generalizada de las funcionalidades que se implementarán y un resumen global del funcionamiento de la aplicación, con el fin de que tenga mayor información y facilitar el seguimiento de la misma. Se describen las líneas que va a seguir la aplicación a desarrollar, explicando a grandes rasgos las funciones con las que va a contar, y el funciona-

miento de las mismas. Además se propondrán futuras opciones de mejora de la aplicación así como la posibilidad de añadirla a otros sistemas software de mayor envergadura, con el fin de ampliar o mejorar sus funcionalidades.

Ámbito del Sistema

La aplicación a desarrollar es un sistema de análisis del movimiento en secuencias de vídeo mediante técnicas basadas en flujo óptico. Se trata de un sistema independiente, de nueva creación, sin relación con ninguna otra aplicación, lo cual no impide que pueda relacionarse con ellas en un futuro para mejorar y aumentar las prestaciones que ésta ofrece. Por lo tanto, no tendrá ningún sistema superior a la misma y en consecuencia este documento no dependerá de ninguno superior.

2.1.2. Descripción General

Se presenta el producto y se define el marco sobre el que la aplicación se desarrollará, identificando sus usuarios, las restricciones impuestas a los programadores y las dependencias a las que está sometida esta especificación.

Funciones del producto

La aplicación se divide en 4 módulos: entrada, interfaz gráfica, análisis y salida. Los módulos, con las funcionalidades correspondientes, son los siguientes:

El módulo de entrada se encarga de la lectura de los datos de entrada y de mostrarlos al usuario. La función será la selección del archivo de vídeo, que contiene la secuencia de imágenes con el movimiento de la escena, objeto del análisis.

La interfaz gráfica comprende todo lo relativo al aspecto visual de la aplicación, encargándose de la interacción directa con el usuario. A través de él se especificarán los parámetros necesarios para el análisis. Las funciones serán las siguientes:

- Visualización del vídeo seleccionado.
- Selección de imagen inicial a analizar o bien análisis automático (análisis completo del vídeo).
- Selección del algoritmo de procesamiento del vídeo, para la obtención del flujo óptico.

El módulo de análisis lleva a cabo el proceso algorítmico para detectar movimiento mediante el método elegido por el usuario y proporcionado por la aplicación. La función es la obtención del flujo óptico según los parámetros previamente seleccionados.

Finalmente, el módulo de salida gestionará los datos obtenidos tras el análisis. La función será la visualización del flujo óptico obtenido.

La descripción detallada de cada una de las funciones se puede encontrar en el apéndice A.

Características de los Usuarios

El usuario final no requiere ningún requisito de formación en informática, siendo no obstante recomendable ciertos conocimientos de visión artificial para la selección adecuada de las imágenes de entrada, la elección de algoritmos e interpretación de los resultados, así como formación básica matemática para la compresión de las matrices finales obtenidas.

Restricciones

Una de las restricciones principales es el formato de los datos de entrada, pues no será posible aceptar todas las extensiones de vídeo disponibles. Los formatos aceptados en esta primera versión serán avi, mp4, m4v, MOV y mpg.

La aplicación será multiplataforma, por lo que no habrá limitaciones, al menos en cuanto a hardware.

Con respecto al lenguaje de programación, se ha decidido usar Matlab al estar más orientado hacia las matemáticas, en especial al tratamiento de matrices, que otros lenguajes y por la previa disponibilidad de los algoritmos de análisis que utilizará la aplicación.

Suposiciones y Dependencias

Al proponerse un diseño multiplataforma, en el supuesto de que haya cambios, por ejemplo en el sistema operativo, los requisitos no se verán afectados.

Si se deseara utilizar otro formato de vídeo para la entrada, sería necesario modificar el módulo de entrada y las funciones que se encarguen de leer datos y transformarlos en matrices para su uso en la aplicación.

En caso de tener necesidad de cambiar cualquier otro requisito, dicha modificación se reflejará en este documento, así como los cambios derivados de éste.

En todos estos casos se actualizará el SRS y el Plan del Proyecto cuando sea necesario.

Requisitos Futuros

Una opción de mejora sería ampliar el ámbito de uso de la aplicación proporcionando más funcionalidades y más concretamente habilitando la posibilidad de aceptar todo tipo de formatos de vídeo como entrada a la aplicación. En este caso el tratamiento de los datos de entrada sería más complejo. Podría también incorporarse la posibilidad de lectura de datos de vídeos en tiempo real, por lo que la aplicación tendría que estar aceptando vídeo de entrada de forma continua.

Si se quisiera añadir otros algoritmos, bastaría con proporcionar sus códigos e incorporar en la interfaz las opciones pertinentes para su uso, así como la entrada de los parámetros necesarios.

Por último, podría agregarse la opción de guardar las salidas obtenidas en archivos externos, para un futuro estudio por parte del usuario.

2.1.3. Requisitos específicos

Describe de forma general los requisitos de funcionamiento del producto así como los factores, restricciones y dependencias que afectan a su desarrollo. La aplicación consta esencialmente de dos partes bien diferenciadas, que origina los correspondientes requisitos, a saber: interfaz de usuario, para entrada/salida de datos e información y funcionalidades específicas de la aplicación.

Interfaces externas

La interfaz de usuario permitirá seleccionar los datos de entrada, residentes en disco. Una vez seleccionados y comprobado que son correctos, informa al usuario. En caso contrario, se le dirigirá a la pantalla de análisis. Aquí podrá seleccionar la imagen inicial (la imagen final se selecciona automáticamente) a analizar o bien realizar una selección automática (esto es, el vídeo completo), el algoritmo, los parámetros en el caso del algoritmo de Gauss-Seidel [7], y dar comienzo al análisis. Además se le permitirá seleccionar un nuevo archivo de entrada, sustituyendo al anterior.

Restricciones de Diseño

Los recursos hardware utilizados por el software son bastante bajos. En cuanto al rendimiento, la aplicación se desarrolla con el fin de que sólo pueda ser accesible por parte de un único usuario. Además, habrá un único archivo de vídeo abierto en cada momento.

Atributos del Sistema

A la aplicación a desarrollar se la presupone un elevado nivel de estabilidad y fiabilidad, persiguiendo el objetivo de crear una aplicación sencilla de uso y minimizar el número de errores y problemas que ésta pueda generar una vez instalada y durante su utilización por parte del cliente.

No hay restricciones de seguridad en cuanto al tipo de usuarios de la aplicación, pues no se solicita ningún tipo de cuenta ni contraseña, existiendo un único perfil de usuario.

En cuanto a la fiabilidad, el sistema evitará que se introduzca información fallida durante el funcionamiento de la aplicación.

2.2. Plan del Proyecto

El Plan del Proyecto es el documento principal para la gestión del software del proyecto. Describe los procesos técnicos y de gestión necesarios para desarrollar productos software que satisfagan los requisitos del proyecto.

2.2.1. Introducción

Este apartado proporciona una visión general del propósito, objetivos y restricciones del proyecto.

Propósito y objetivos

El propósito del proyecto se ha presentado en el capítulo 1.2, así como los principales usos de la aplicación.

Supuestos y restricciones

Se usan algoritmos previamente implementados. Al tratarse de secuencias de imágenes consecutivas en el tiempo se puede suponer la ausencia de cambios bruscos en el brillo y luminosidad, si bien se asumen pequeñas variaciones de intensidad debido al comportamiento del propio sensor y a las variaciones que surgen como consecuencia de movimientos poco relevantes tales como movimiento de hojas de árboles por pequeñas ráfagas de viento o similares.

Resumen de planificación

El proyecto se encuentra dividido en dos fases. La primera fase incluye:

- La comunicación con el cliente, donde se acuerdan los términos del proyecto.
- El análisis de los requisitos.
- La planificación del proyecto, que da lugar al presente plan.

La segunda fase consta de la generación de código y pruebas, de forma que el código se estructura en módulos de acuerdo a su funcionalidad. Sobre cada módulo se aplica la correspondiente fase de pruebas.

Evolución del plan

El plan se actualiza semanalmente, de acuerdo con las necesidades surgidas. Durante el periodo de implementación del software se trabaja actualizando la aplicación por módulos, esto es, cuando se finaliza un módulo se genera una copia de seguridad de lo ya implementado y se actualiza con el nuevo módulo. Las actualizaciones no programadas se reflejan en el plan del proyecto, comprobando que cumplen los requisitos especificados, así como los cambios en la planificación.

2.2.2. Planes de proceso de gestión

Esta sección especifica los procesos de gestión del proyecto, incluyendo el plan de estimación, el plan de adquisición de recursos, el plan de formación, el plan de trabajo, el plan de control y el plan de gestión de riesgos.

Plan de estimación

Dentro del plan de estimación se distingue, por una parte, la estimación del tamaño del proyecto. Para ello se ha usado una estimación basada en el problema orientada a los puntos de función. No se realiza una estimación orientada a las líneas de código puesto que no se considera relevante en este proyecto, principalmente porque para la interfaz gráfica se usan herramientas automáticas como guide, proporcionada por Matlab [2]. Además se utilizan algoritmos previamente implementados.

Los puntos de función proporcionan un enfoque indirecto, estimando las características del dominio de información: entradas externas, salidas externas, consultas externas, archivos lógicos internos y archivos de interfaz externos.

Por la falta de datos históricos, la estimación de los valores optimista, probable y pesimista del tamaño se basa en la intuición. El factor de ponderación de complejidad es el promedio y el valor esperado se calcula según la siguiente fórmula:

$$VE = \frac{VO + 4VPR + VPE}{6} \tag{2.1}$$

El valor de los puntos de función, una vez obtenido el total de los valores esperados ponderados por sus respectivos pesos, se obtiene a partir de

$$PF = total * (0{,}65 + 0{,}01 * \sum_{i=1}^{14} F_i) \tag{2.2}$$

Desde un punto de vista teórico (no siendo relevante en este proyecto por su tamaño y por tener un único integrante) el esfuerzo total, siendo Pr la productividad organizacional promedio, viene dado por:

$$E = \frac{PF}{Pr} \tag{2.3}$$

Los detalles y resultados de este proceso se encuentran en el apéndice B.

Por otra parte se tiene la estimación de recursos. El software reutilizable disponible para el proyecto son los algoritmos de Lucas-Kanade y Gauss-Seidel previamente implementados, tomados del material adicional de [7]. Las herramientas software y hardware que se utilizan se describen más adelante en la sección 2.2.2.

Plan de adquisición de recursos

La tabla 2.1 recoge los recursos utilizados a lo largo del proyecto, junto con su descripción y las fases en las que se usa, es decir, las actividades de trabajo (definidas en 2.2.1) que requieren el uso de dichos recursos.

Recursos	Descripción	Fase
Ordenadores	Personales y localizados en los laboratorios de la facultad de informática de la UCM.	Durante todo el proyecto.
Estándar IEEE 1058	Índice propuesto para llevar a cabo el plan de proyecto.	Planificación del proyecto.
Estándar IEEE 830	Índice propuesto para llevar a cabo la especificación de requisitos.	Especificación de requisitos.
MS Project Professional 2016	Programa informático empleado para la planificación del proyecto.	Planificación del proyecto.
Google Docs	Paquete de programas informáticos online (hojas de cálculo, editores de texto?).	Planificación y análisis de requisitos.
Matlab R2015a	Programa informático empleado para la creación del código en lenguaje Matlab.	Generación de código y pruebas.
Biblioteca de la UCM	Localización desde donde se lleva a cabo el desarrollo del proyecto.	Durante todo el proyecto.
Gliffy [13]	Herramienta online para creación de diagramas.	Generación de código y pruebas.

Tabla 2.1: Recursos del proyecto.

Plan de formación

Se han aprovechado los conocimientos adquiridos en las siguientes dos materias: Tecnología de la Programación e Ingeniería del Software. Además se han realizado consultas en [8], [9], y en internet, principalmente para la programación en Matlab. Respecto a esta última, se ha utilizado la experiencia adquirida durante las asignaturas de Métodos Numéricos y Análisis Numérico. En cuanto al

desarrollo de los algoritmos, se ha utilizado el libro de referencia [7], junto con el material digital adicional que lo acompaña, así como el libro de problemas del mismo nombre [10].

Por último también se lleva a cabo una formación externa de forma autodidacta con el fin de ampliar conocimientos en los campos anteriormente citados.

Plan de trabajo

Las actividades de trabajo, con su planificación temporal, y distribuidas en las fases descritas en 2.2.1, son las siguientes:

Nombre de tarea	Duración	Comienzo	Fin
FASE 1	**66 días**	**lun 02/11/15**	**lun 01/02/16**
Com. Cliente	21 días	lun 02/11/15	lun 30/11/15
Análisis de requisitos	31 días	lun 23/11/15	lun 04/01/16
Planificación del proyecto	31 días	lun 21/12/15	lun 01/02/16
Fin fase 1	0 días	lun 01/02/16	lun 01/02/16
FASE 2	**75 días**	**lun 15/02/16**	**vie 27/05/16**
M. Entrada	**20 días**	**lun 15/02/16**	**vie 11/03/16**
Generación del código	10 días	lun 15/02/16	vie 26/02/16
Pruebas del softwa	10 días	lun 29/02/16	vie 11/03/16
M. Interfaz Gráfica	**30 días**	**lun 14/03/16**	**vie 22/04/16**
Generación del código	25 días	lun 14/03/16	vie 15/04/16
Pruebas del software	5 días	lun 18/04/16	vie 22/04/16
M. Análisis	**15 días**	**lun 25/04/16**	**vie 13/05/16**
Generación del código	10 días	lun 25/04/16	vie 06/05/16
Pruebas del software	5 días	lun 09/05/16	vie 13/05/16
M. Salida	**10 días**	**lun 16/05/16**	**vie 27/05/16**
Generación del código	5 días	lun 16/05/16	vie 20/05/16
Pruebas del software	5 días	lun 23/05/16	vie 27/05/16
Fin fase 2	0 días	vie 27/05/16	vie 27/05/16

Figura 2.1: Actividades de trabajo

El calendario de actividades de trabajo de la figura 2.2 muestra el desarrollo de las tareas en el tiempo, esto es, la planificación temporal del proyecto.

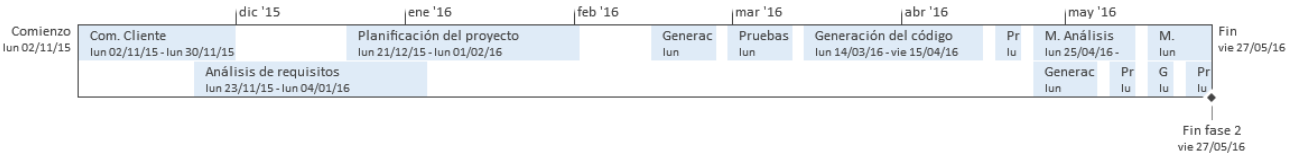


Figura 2.2: Calendario de las actividades de trabajo

En el diagrama de Gantt mostrado en la figura 2.3 se encuentra representado quién es el predecesor y el sucesor, de haberlos, de cada actividad de trabajo. Resume todas las tareas implicadas en el proyecto, detalladas previamente en la figura 2.1, y su orden y situación temporal.

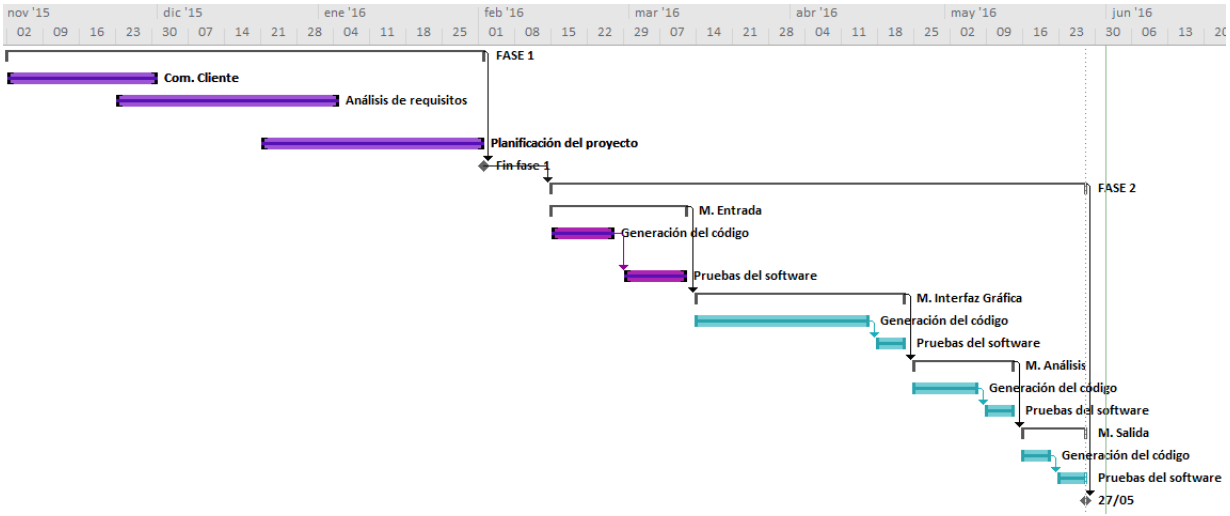


Figura 2.3: Diagrama de Gantt

El camino crítico mostrado en la figura 2.4 está formado por las tareas críticas, esto es, aquellas cuyo retraso provoca el retraso del final del proyecto. Por tanto, el camino crítico es aquel de mayor duración. En este proyecto, al ser la planificación prácticamente lineal, el camino crítico está formado por todas las tareas de la segunda fase.

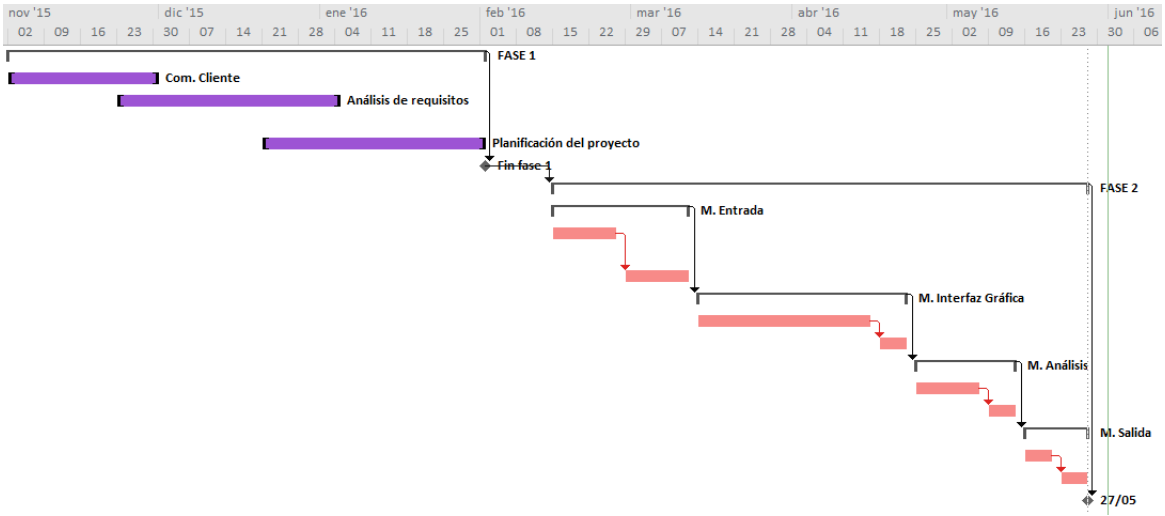


Figura 2.4: Camino crítico

El diagrama de Pert (Program Evaluation and Review Technique) es una variación del método del camino crítico que analiza las tareas involucradas en completar el proyecto, especialmente el tiempo para completar cada tarea, e identifica el tiempo mínimo necesario para completar el proyecto total. Se trata de un diagrama de red que incluye información sobre precedencia de tareas de una forma más clara que el diagrama de Gantt de la figura 2.3.

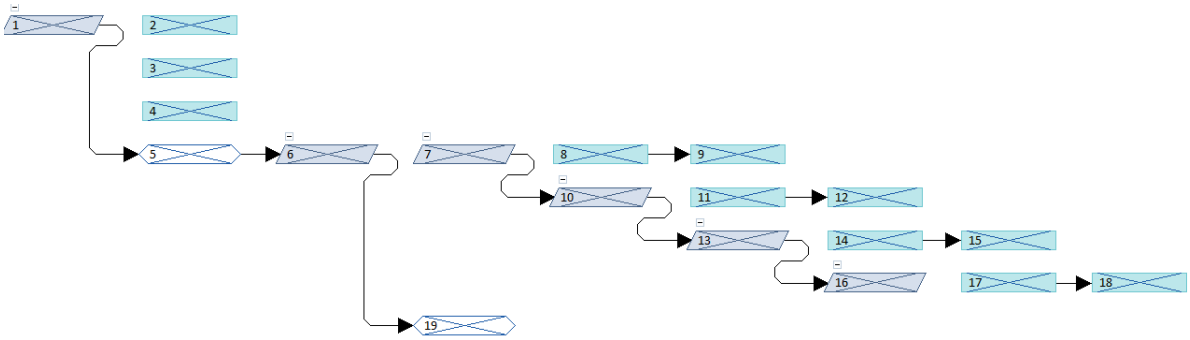


Figura 2.5: Diagrama de Pert

Plan de control de requisitos

El propio desarrollo del proyecto implica su actualización a medida que se completa el mismo. La actualización abarca el propio documento, así como la SRS.

Respecto al control de cambios en los requisitos, en 2.1.2 quedan reflejadas las consecuencias y las medidas que se llevarían a cabo para solucionarlas.

Plan de control de planificación

Para comparar el progreso real con los avances previstos se utilizan medidas básicas definidas por el análisis del valor ganado:

- CPTP: Coste Presupuestado del Trabajo Planificado.
- CPTR: Coste Presupuestado del Trabajo Realizado.
- VP (Variación de la planificación) = CPTR-CPTP.
- IRP (Índice de Rendimiento de la Planificación) = CPTR/CPTP.

Tanto el CPTP como el CPTR vienen dados por el archivo de Microsoft Project como el porcentaje completado de las actividades. Esto es, con un 0 %, 25 %, 50 %, 75 %, o el 100 %. Se van actualizando a lo largo del proyecto.

La VP sirve para comprobar si el progreso es bueno o no, viendo si la variación de planificación es positiva (VP ≥ 0). Además se espera que el IRP sea mayor que uno (IRP ≥ 1).

Análisis y gestión de riesgos

Se sigue el modelo de Boehm, con ayuda de formularios de gestión de riesgos para tratarlos con la mayor eficiencia posible, siguiendo una estrategia reactiva, pues en lugar de tratar de prever los riesgos potenciales antes de que ocurran, se realizan las tareas asignadas y, en caso de encontrarse un riesgo, se evalúa y resuelve lo más rápido posible.

Los problemas se clasifican en:

- Bajo (o tolerable). El problema es tan pequeño que la ruta crítica no se verá afectada.
- Medio. Problema que tiene un impacto a la hora de programar pero no se espera que afecte a la ruta critica.
- Alto. Problema que afecta al tiempo pero no directamente en la ruta crítica. Si no se resuelve a tiempo podría afectar a esta.
- Intolerable. El problema afecta al tiempo de entrega de las actividades en la ruta crítica.

Las probabilidades de ocurrencia del riesgo se clasifican en frecuente, probable, ocasional, remota e improbable. El impacto, descrito en [8], puede ser catastrófico, crítico, serio, menor o despreciable. Para obtener el nivel de riesgo, se utiliza la tabla SQAS-SEI [12].

A continuación se exponen los riesgos más amenazadores, identificados durante las etapas de desarrollo de la presente aplicación, realizando el correspondiente estudio sobre ellos:

1. Planificación demasiado optimista. Este riesgo es ocasional, pues el plan del proyecto se actualizará ante los cambios surgidos, y su impacto serio, pues la fecha final de entrega no es prorrogable. El nivel de riesgo es medio.

2. Riesgos técnicos de implementación, ya que hasta que no se implementa el código no se sabe lo difícil que puede ser construirlo u optimizarlo. Este riesgo es probable y su impacto serio, pudiendo producir grandes retrasos. El nivel de riesgo es alto.

3. Problemas en cuanto a la eficiencia del código, esto es, que la aplicación tarde en responder más de lo esperado. El riesgo será remoto y su impacto menor ya que los algoritmos de análisis del movimiento se encuentran ya desarrollados y optimizados. Sí podría ocurrir durante el procesamiento de las imágenes en la selección manual, pero al tratarse solamente de dos, el retraso sería prácticamente imperceptible. En el análisis automático, sin embargo, supondría un mayor problema, pues todos los fotogramas son analizados. El nivel de riesgo es bajo.

4. Problemas con la interfaz gráfica de la aplicación, debido a la inexperiencia de los programadores en este campo. Su probabilidad es ocasional y el impacto será crítico debido a que la interfaz es esencial en esta aplicación. El nivel de riesgo es alto.

5. Continua corriente de cambios en los requisitos. El riesgo será probable y el impacto serio ya que un cambio en las últimas etapas podría ser desastroso. Podría conllevar grandes modificaciones en el código. El nivel de riesgo es alto.

6. Fallos en los sistemas informáticos, tales como averías, cuelgues... La probabilidad es remota y su impacto seria menor , teniendo en cuenta que el proyecto es a largo plazo, por lo que retrasos puntuales no tendrían gran efecto.El nivel de riesgo es bajo.

7. El erróneo desarrollo de la interfaz de usuario, esto es, que el cliente final encuentre poco atractiva la interfaz gráfica con la que se presenta la aplicación, o bien que no cumpla sus expectativas. La probabilidad es ocasional, pues para ello se tiene la especificación de requisitos, y su impacto serio ya que podría dar lugar a muchos cambios de implementación, lo que retrasaría la entrega del proyecto. El nivel de riesgo es medio.

8. La planificación no incluye tareas necesarias La probabilidad es ocasional y su impacto sería catastrófico ya que podría conllevar retrasos importantes. El nivel de riesgo es intolerable.

9. Bajas del personal ocasionadas por motivos personales, problemas familiares, etc. Es improbable que esto ocurra pero su impacto sería catastrófico por haber una única persona en el equipo de trabajo. El nivel de riesgo es medio.

2.2.3. Planes de proceso técnico

Este apartado especifica el modelo de desarrollo de proceso y los planes para establecer y mantener la estructura del proyecto.

Modelo de proceso

La aplicación o producto final se ha desarrollado bajo el modelo de proceso en cascada o *waterfall*, con la variante de Sommerville desarrollada en su libro [9]. Se trata de un enfoque sistemático y secuencial para el desarrollo del software, que comienza con la especificación de los requisitos por parte del cliente y avanza a través de planificación, análisis de los requisitos, diseño del software y generación del código, para concluir con las pruebas de software. Este modelo queda ilustrado en la figura 2.6.

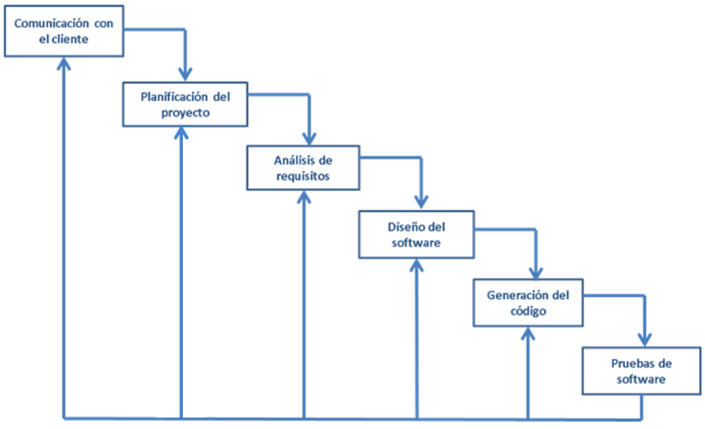


Figura 2.6: Modelo de proceso en cascada

Plan de infraestructura

El código generado se almacena a lo largo del proyecto en disco. Además, se realiza una copia que es guardada en una cuenta en Dropbox. Los datos se actualizan después de cada sesión de trabajo.

Para llevar a cabo el desarrollo de la aplicación ha sido necesario la instalación de un entorno de desarrollo (Matlab R2015a) en Ubuntu 14.0, así como Microsoft Project Professional 2016 [11] en Windows 10.

2.2.4. Planes de soporte de proceso

Esta sección describe los planes de soporte para los distintos procesos que tienen lugar a lo largo del desarrollo del proyecto. Incluye los planes para la gestión de la configuración, de garantía y calidad, de resolución de problemas y de mejoras del proceso.

Plan de gestión de la configuración

El objetivo de la gestión de la configuración es mantener la integridad de los productos que se obtienen a lo largo del desarrollo del proyecto, garantizando que no se realizan cambios incontrolados, a través del control y registro de estos cambios.

Para aplicar el plan de gestión al proyecto desarrollado, en primer lugar se han seleccionado los ítems de configuración, esto es, los elementos cuyo cambio pudiera resultar crítico para el buen desarrollo del proyecto, con el objetivo de asegurar su integridad, evaluándose en todo momento los cambios y decidiendo sobre su implementación.

Por cada elemento se ha generado una línea base y una vez que el producto se ha verificado, probado y validado con el cliente, se obtiene la línea base final. La línea base relativa a cada fase no se ha modificado en ningún caso, y cuando ha sido necesario la modificación ha afectado únicamente a la planificación. Esto queda reflejado en el diagrama de Gantt (ver figura 2.3).

Plan de Garantía de Calidad

El propósito de este plan es vigilar que se cumplan las previsiones hechas para el proyecto, y que éste se vaya a poder entregar en la fecha establecida, así como garantizar el absoluto cumplimiento de las exigencias impuestas previamente durante el transcurso de desarrollo del software, así como su posterior validación y futuro mantenimiento.

Se realizan revisiones de gestión y revisiones técnicas, prestando especial atención a que el proyecto se esté realizando conforme a los requisitos exigidos por el cliente, que se cumplen las planificaciones elaboradas de antemano, y que en están recogidas en el Plan del Proyecto, actualizando la parte que no cumple con los requisitos, o que precise de algún cambio.

Los documentos que deben ser revisados son tanto el plan del proyecto, como la especificación de requisitos, para su comparación con el estado actual del proyecto, así como los informes que se vayan generando de errores durante el proceso de implementación.

Cualquier problema surgido durante las revisiones se recoge formalmente en un informe, que es añadido a la documentación del proyecto, y solventar dicho problema tendrá que ser prioridad máxima, pues puede generar otros problemas mayores.

Cada informe generado durante las revisiones es guardado durante todo el tiempo que dure el proyecto y, una vez terminado, sólo se conservan aquellos informes que contengan problemas graves surgidos durante el desarrollo del software, y que hayan afectado gravemente al proyecto.

Plan de Resolución de Problemas

Los cambios necesarios se estudian según avance el desarrollo de la aplicación, evaluando su riesgo e impacto, ya que es posible que sean modificados la especificación de requisitos y el propio plan del proyecto en caso de ser necesario. Los problemas se resuelven por orden de gravedad.

Finalmente también es necesario actualizar la previsión temporal del proyecto según se vaya avanzando en el desarrollo del mismo utilizando la herramienta destinada a ello (Microsoft Project) con el objetivo de valorar la eficiencia y eficacia al finalizar el proyecto.

Plan de mejoras del proceso

El plan de mejoras se aplica sobre todo durante la fase de generación del código en la cual el programador tiene la suficiente autonomía para realizar cualquier mejora en el ámbito de la lógica.

Cualquier mejora del proceso en el proyecto que pueda beneficiar al rendimiento actual es considerada para la implementación, de manera que las fases del proyecto restantes también mejoren. Los cambios en la organización que dichas mejoras puedan producir deben ser documentados.

En el momento en el que se identifique el proceso o problema a mejorar, hay que encontrar las causas que originan el problema, definir los proyectos y acciones de mejora y planear y dar seguimiento a dichas acciones.

2.3. Diseño de alto nivel

De forma general la aplicación se divide en entrada de datos, selección de opciones, procesamiento y salida. A continuación se presentan sus características a alto nivel, junto con los diagramas de secuencia y casos de uso, que definen el diseño de la aplicación.

La **entrada de datos** comprende la lectura del fichero conteniendo el vídeo que se desea analizar, seleccionado por parte del usuario, y de su transformación para mostrarlo en la interfaz de usuario y para su posterior análisis. Una vez abierto el vídeo, el usuario puede proceder a **seleccionar las opciones** y parámetros requeridos para el análisis por parte de los procedimientos específicos de tratamiento de imágenes y concretamente de los métodos de Lucas-Kanade y Gauss-Seidel. Tras comprobar la corrección de estas opciones, está en disposición de realizar el análisis para la detección de movimiento en el vídeo seleccionado.

El **procesamiento** consiste en el análisis del vídeo, según los parámetros seleccionados, a través de los métodos de Lucas-Kanade y Gauss-Seidel. La explicación detallada de éstos se encuentra en el siguiente capítulo. Además, este procesamiento puede realizarse sobre un par de imágenes, siendo la primera elegida por el usuario, o del vídeo completo. Por último, se muestra la **salida**, esto es, la visualización de los resultados obtenidos.

2.3.1. Diagramas

A continuación se presentan los diagramas concretos, cuya finalidad es aclarar y mostrar de forma gráfica la funcionalidad de la aplicación desarrollada.

El diagrama de secuencia de la aplicación muestra las interacciones entre los objetos en el orden en que éstas ocurren. Recoge lo explicado anteriormente de forma visual.

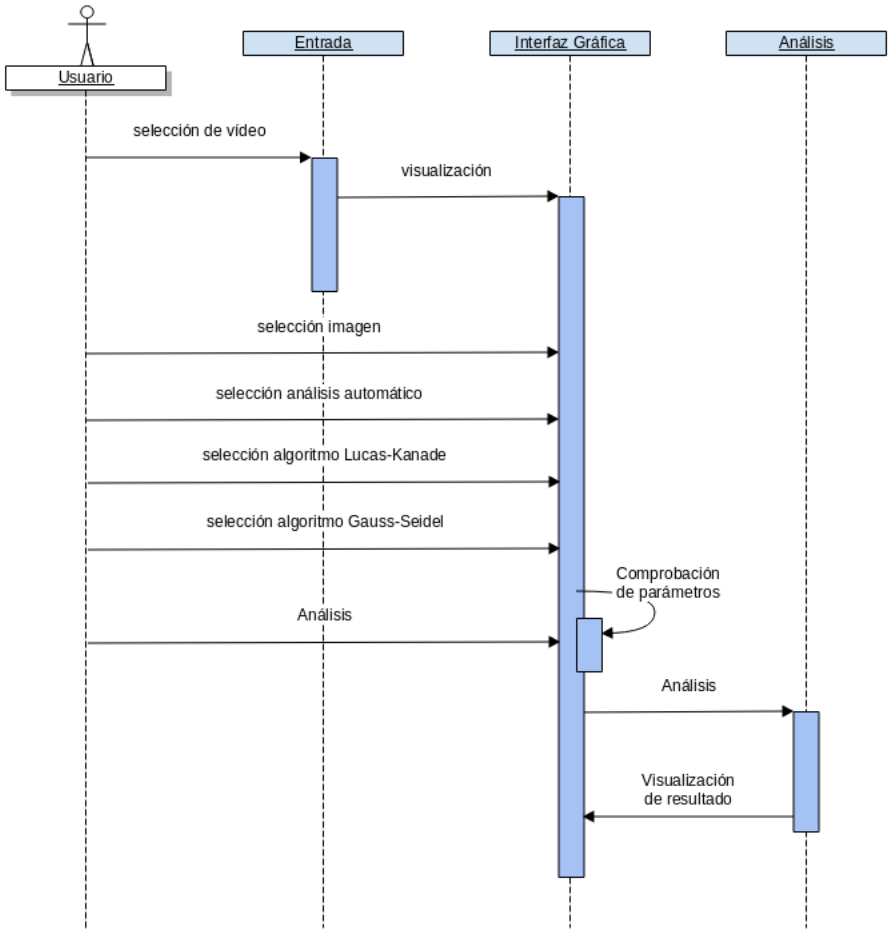


Figura 2.7: Diagrama de secuencia

El diagrama de casos de uso, mostrado en la figura 2.8, es una representación gráfica de las interacciones entre los elementos de un sistema. Identifica, aclara y organiza los requisitos del sistema.

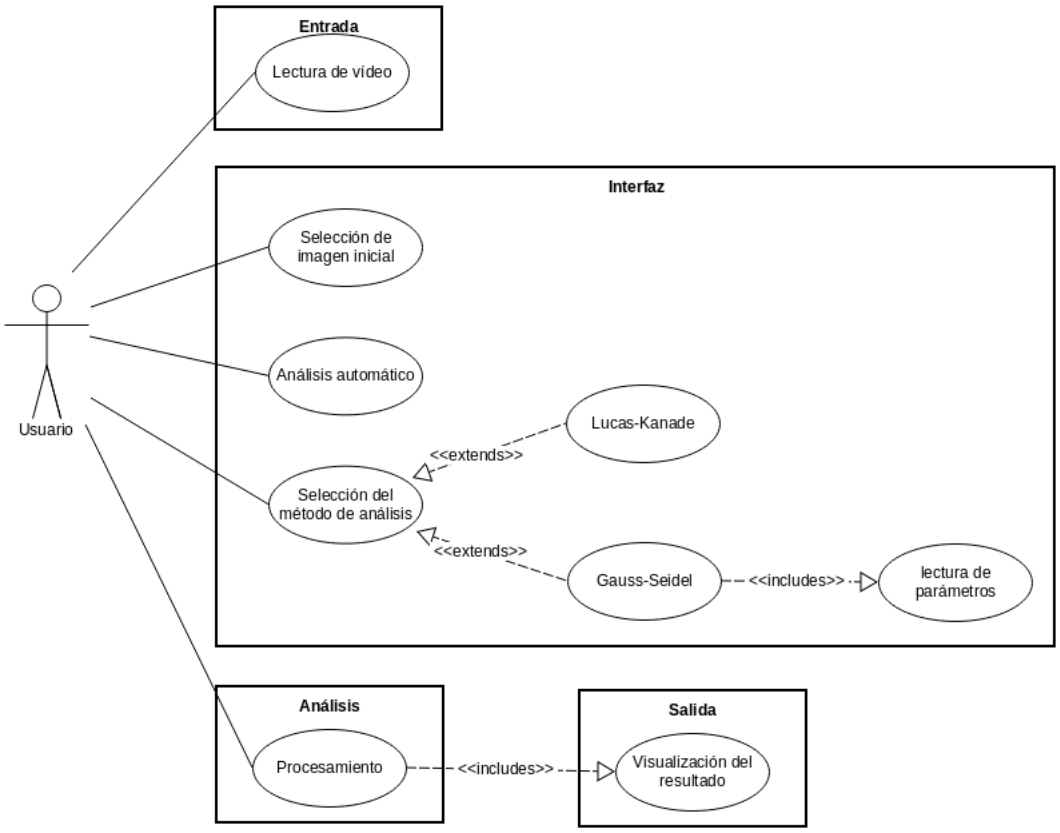


Figura 2.8: Diagrama de casos de uso

Capítulo 3

Descripción de los métodos de análisis del movimiento

El objetivo del análisis realizado por la aplicación es la detección de movimiento en secuencias de vídeo, con fines de vigilancia. Esto se lleva a cabo mediante técnicas específicamente desarrolladas para obtener lo que se conoce como flujo óptico. A continuación se introduce este concepto y los métodos concretos usados en este proyecto: Lucas-Kanade y Gauss-Seidel.

3.1. Flujo óptico

El estudio que se va a realizar es la detección de movimiento, para lo cual se usa una cámara estática que observa continuamente la misma escena, y cuya principal aplicación es el campo de la seguridad. Se realiza un análisis dinámico de imágenes, esto es, imágenes consecutivas de una secuencia, con un intervalo temporal entre ellas pequeño y sin ocurrir cambios importantes entre ellas.

El movimiento, tridimensional, se representa bidimensionalmente a través del campo de movimiento: se asigna un vector velocidad (que es el que se obtendrá durante el análisis) a cada punto, y que representa la dirección y magnitud del movimiento. Así, el flujo óptico refleja los cambios en la imagen debidos precisamente al movimiento de los objetos en la escena.

En escenas reales, sucede que siempre se producen cambios de iluminación que afectan a los niveles de intensidad del mismo punto de la escena en distintas imágenes de la secuencia. Esto ocurre incluso aunque no haya habido un movimiento significativo en la escena desde el punto de vista de la videovigilancia y se debe a pequeñas variaciones de intensidad luminosa en la escena o al propio sensor de captura del vídeo. Debido a esto, es necesario considerar el supuesto de que estas pequeñas variaciones no son significativas desde el punto de vista del análisis del flujo, considerando en este caso que los objetos estáticos en la escena poseen niveles de intensidad constante en las diferentes imágenes de la secuencia, lo que se determina en función de la magnitud del flujo óptico, de suerte que valores de magnitud por debajo de un determinado umbral se asume que no corresponden a movimiento propiamente dicho, considerándose en tal caso valores de intensidad constante.

El objetivo del análisis es estimar el movimiento relativo entre los objetos presentes en la escena y las imágenes. La observación de los cambios entre los niveles de intensidad de las imágenes se basa en el gradiente espacial en el punto (x, y), y para ello se parte de la siguiente ecuación,

$$\frac{\partial f}{\partial t} + v.\nabla f = 0 \tag{3.1}$$

donde f es la función de intensidad de la imagen.

Además, siendo $f\,(x, y, t)$ la función de intensidad del píxel (x, y) en el instante t, se tiene que

$$\begin{aligned} f\,(x+dx, y+dy, t+dt) &= f\,(x, y, t) + \frac{\partial f}{\partial x}dx + \frac{\partial f}{\partial y}dy + \frac{\partial f}{\partial t}dt + O\left(\partial^2\right) \\ &= f\,(x, y, t) + f_x dx + f_y dy + f_t dt + O\left(\partial^2\right) \end{aligned} \tag{3.2}$$

es un desarrollo en serie de Taylor de primer orden para la variación espacio temporal de la intensidad de la imagen (x, y).

Para dx, dy, dt muy pequeños el resto del desarrollo de Taylor es despreciable, y además se puede asumir que la vecindad inmediata de (x, y) se traslada una pequeña distancia durante dt, por lo que la ecuación 3.1 se transforma en

$$-f_t = f_x \frac{dx}{dt} + f_y \frac{dy}{dt} \tag{3.3}$$

Así, se busca obtener la velocidad, definida como $v = (\frac{dx}{dt}, \frac{dy}{dt})$. Puede observarse que esta última ecuación es equivalente a 3.1. Para asegurar una intensidad constante, según el razonamiento realizado previamente, se añade la restricción de gradiente constante a lo largo de la trayectoria del movimiento.

3.2. Método de Lucas-Kanade

Se trata de un método local para estimar el movimiento basado en la observación del cambio de los niveles de intensidad en la imagen, como ya se ha explicado en la sección anterior.

Este método expresa la ecuación 3.3 de la siguiente forma:

$$\begin{bmatrix} \partial^2 f/\partial x^2 & \partial^2 f/\partial x \partial y \\ \partial^2 f/\partial x \partial y & \partial^2 f/\partial y^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = -\frac{\partial \left(\overrightarrow{\nabla f}\right)}{\partial t} \tag{3.4}$$

Siendo Ω un entorno de vecindad alrededor de cada punto, se puede expresar como

$$\begin{bmatrix} \sum_\Omega f_x^2 & \sum_\Omega f_x f_y \\ \sum_\Omega f_x f_y & \sum_\Omega f_y^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = -\begin{bmatrix} \sum_\Omega f_t f_x \\ \sum_\Omega f_t f_y \end{bmatrix} \equiv Av = b \tag{3.5}$$

Es decir, el método consiste en obtener la solución de la ecuación $Av = b$, que representa la igualdad anterior.

$$v = \left(A^T A\right)^{-1} A^T b \tag{3.6}$$

3.3. Método de Gauss-Seidel

Este método, que a diferencia del anterior es un método global, surge a raíz del problema de apertura. Cuando solo se dispone de una ventana de visibilidad alrededor de un punto, y el movimiento de la cámara provoca un cambio en la posición del contorno, es difícil determinar la dirección del movimiento. Únicamente se puede obtener la componente ortogonal del vector velocidad (v^0). Así, se plantea la siguiente ecuación análoga a 3.1:

$$v.\frac{\nabla f}{\|\nabla f\|} = -\frac{f_t}{\|\nabla f\|} \tag{3.7}$$

El vector velocidad puede descomponerse en la suma de dos componentes, v^0 la componente ortogonal y v^t la componente tangencial. Si el contorno posee intensidad constante, v^t es nula porque no hay variación tangencial de intensidad a lo largo del contorno. No así cuando se atraviesa el contorno que sí la hay. Por lo tanto, se tiene para cada punto de la imagen la siguiente ecuación

$$\|v^0\| = -\frac{f_t}{\|\nabla f\|} \tag{3.8}$$

Ante este problema, se añade a lo anterior la restricción de continuidad de la velocidad, esto es, que los puntos próximos entre sí en el plano de la imagen se mueven de forma similar. Esta técnica consiste en minimizar el error al cuadrado:

$$E^2(x,y) = (f_x u + f_y v + f_t)^2 + \lambda\left(u_x^2 + u_y{}^2 + v_x^2 + v_y^2\right) \tag{3.9}$$

El primer sumando es una solución de la ecuación 3.3 y el segundo representa la restricción de continuidad de la velocidad que hemos introducido, siendo λ un multiplicador de Lagrange. Dicho multiplicador pondera la continuidad de la velocidad y será mayor cuanto menor ruido haya (medidas de intensidad constantes, que no dependen del sensor). El ruido lo suele introducir el sensor, de forma que si es de baja calidad mayor ruido introducirá.

El método de Gauss-Seidel consiste en resolver las siguientes ecuaciones diferenciales, obtenidas usando técnicas basadas en el cálculo de variaciones:

$$\begin{aligned}\left(\lambda^2 + f_x^2\right) u + f_x f_y v &= \lambda^2 \bar{u} - f_x f_t \\ f_x f_y u + \left(\lambda^2 + f_y^2\right) v &= \lambda^2 \bar{v} - f_y f_t\end{aligned} \tag{3.10}$$

donde $\bar{u}$,$\bar{v}$ son valores medios de la velocidad en las direcciones x e y en alguna vecindad de (x,y). La solución de 3.10 viene dada por

$$\begin{aligned}u &= \bar{u} - f_x \frac{P}{D} \\ v &= \bar{v} - f_y \frac{P}{D}\end{aligned} \tag{3.11}$$

con

$$P = f_x \bar{u} + f_y \bar{v} + f_t, \qquad D = \lambda^2 + f_x^2 + f_y^2 \tag{3.12}$$

Así, se obtiene el flujo óptico de forma iterativa a partir de pares de imágenes dinámicas consecutivas. Los pasos son los siguientes:

1. Inicializar todos los vectores de velocidad a 0 y el número de iteración k a 0.

2. Sea k el número de iteración. Obtener los valores u^k, v^k para todos los píxeles con coordenadas (x, y).

$$\begin{aligned} u^{k+1}(x,y) &= \bar{u}^k(x,y) - f_x(x,y)\frac{P(x,y)}{D(x,y)} \\ v^{k+1}(x,y) &= \bar{v}^k(x,y) - f_y(x,y)\frac{P(x,y)}{D(x,y)} \end{aligned} \tag{3.13}$$

3. Parar si $\sum_x \sum_y E^2(x,y) < \varepsilon$, siendo ε el máximo error permitido. En cualquier otro caso, ir al paso 2.

Este método puede ampliarse a una serie de imágenes dinámicas consecutivas. Basta con tomar como solución inicial la solución final de la imagen previa. Esto se usará durante el análisis automático de la secuencia de imágenes completa.

La complejidad de este algoritmo es de $O(n^p)$, donde p expresa el orden de la ecuación diferencial 3.10. Se obtiene un error razonable con las primeras 10-20 iteraciones.

Capítulo 4

Implementación y resultados

Este capítulo describe las herramientas y recursos utilizados para el desarrollo de la aplicación, además de mostrar una serie de resultados representativos mediante ejemplos de imágenes analizadas junto con las correspondientes interfaces de usuario.

4.1. Implementación

El lenguaje de programación para el desarrollo del código ha sido Matlab [1], en particular la versión R2015a. Para el desarrollo de la interfaz se ha utilizado la herramienta guide [2], el entorno interactivo de desarrollo gráfico de interfaz de usuario de Matlab.

Se ha desarrollado una interfaz gráfica amigable, que facilita la interacción con el usuario. Para la entrada de datos se ha utilizado la rutina de Matlab de lectura de ficheros, presentando al usuario un cuadro de diálogo de exploración de carpetas, que muestra por defecto los archivos de extensión *mp4*. Tras la lectura del archivo de vídeo, éste se muestra en la interfaz para ser visualizado por parte del usuario y para la elección del fotograma de referencia. Es bien sabido que el análisis del movimiento mediante los métodos de Lucas-Kanade o Gauss-Seidel requiere una imagen de referencia sobre la que se determinan las variaciones espacio-temporales que aparecen en otra imagen de la secuencia, que

bien puede ser posterior o anterior en el tiempo respecto de la de referencia. En la implementación realizada se fija la imagen de referencia como la que aparece en pantalla, siendo la siguiente en el tiempo la que se utiliza para la detección del movimiento.

La interfaz, además de mostrar el vídeo, presenta todas las opciones disponibles para el análisis. En primer lugar, la selección de análisis automático, esto es, realizar el análisis del vídeo completo. En caso contrario, se selecciona el fotograma a través de la barra de desplazamiento del vídeo. Se puede elegir entre uno de los dos métodos implementados. En el caso de Gauss-Seidel, se pide al usuario introducir los parámetros necesarios y no se le permite realizar el análisis hasta que éstos se han introducido.

Los dos algoritmos implementados se ejecutan de forma automática, mostrando el resultado al terminar. El gráfico obtenido se presenta en una nueva ventana, permitiendo al usuario interaccionar sobre ella, ampliando los resultados para su mejor visualización. En el caso del método de Gauss-Seidel se muestra el flujo óptico obtenido superpuesto sobre la imagen final. En cambio, en el caso de Lucas-Kanade se muestra únicamente el flujo óptico, pues la matriz obtenida por este algoritmo es de distinta dimensión que la imagen.

La aplicación final se proporciona en un ejecutable, por lo que será multiplataforma. Para generarlo se ha utilizado la herramienta de Matlab que permite compartir programas en este lenguaje como aplicaciones autónomas: Matlab Compiler [3]. El usuario final no necesita disponer de la aplicación de Matlab, pero sí se requiere la instalación de Matlab Runtime [4].

4.2. Resultados

A continuación se presentan sendos ejemplos relativos a distintas ejecuciones de la aplicación con vídeos de ejemplo.

4.2.1. Ejemplo 1

En un primer ejemplo, el movimiento en la escena está determinado por el perro desplazándose en la escena. Si bien, a pesar de la existencia de dicho desplazamiento, también se producen movimientos significativos en dos partes relevantes del animal, exactamente en la parte posterior y en la cabeza. Estos movimientos son suficientemente significativos presentándose en sus partes fronterizas con respecto a la escena, y además se producen en todas las direcciones espaciales, no sólo en la dirección principal del movimiento. Estas circunstancias aparecen reflejadas en los resultados que se presentan posteriormente.

Las imágenes a analizar se muestran en la figura 4.1, siendo la imagen final seleccionada automáticamente por la aplicación.

(a) Imagen inicial (b) Siguiente imagen en la secuencia

Figura 4.1: Intervalo de imágenes a analizar

Tras aplicar el método de Lucas-Kanade, el resultado obtenido es el siguiente:

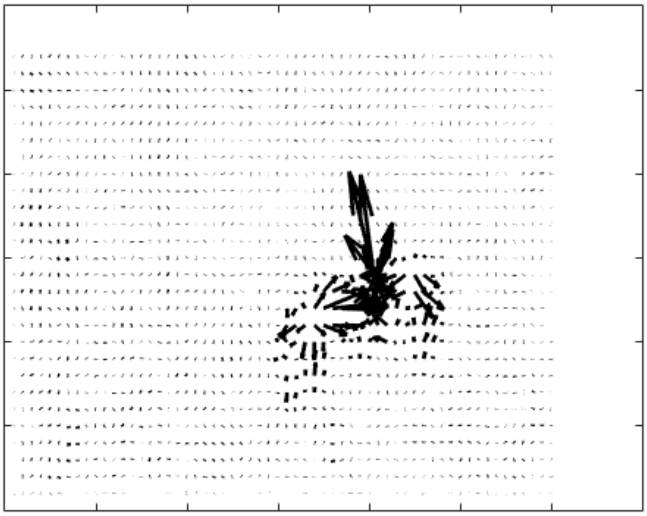


Figura 4.2: Flujo óptico obtenido por el método de Lucas-Kanade

Analizando los mismos fotogramas, pero en este caso con el método de Gauss-Seidel con parámetros $k = 3$, $\varepsilon = 10^{-}8$ y $\lambda = 1$, se obtiene el resultado del cómputo del flujo óptico mostrado en la figura 4.3. En este caso es fácil determinar cómo se localiza el movimiento en las mismas partes significativas de la imagen y bajo los mismos planteamientos anteriores respecto del movimiento en las zonas fronterizas del perro en movimiento. No obstante, ahora en dichas partes aparecen componentes significativas orientadas hacia la dirección principal del movimiento, dada la naturaleza global del método.

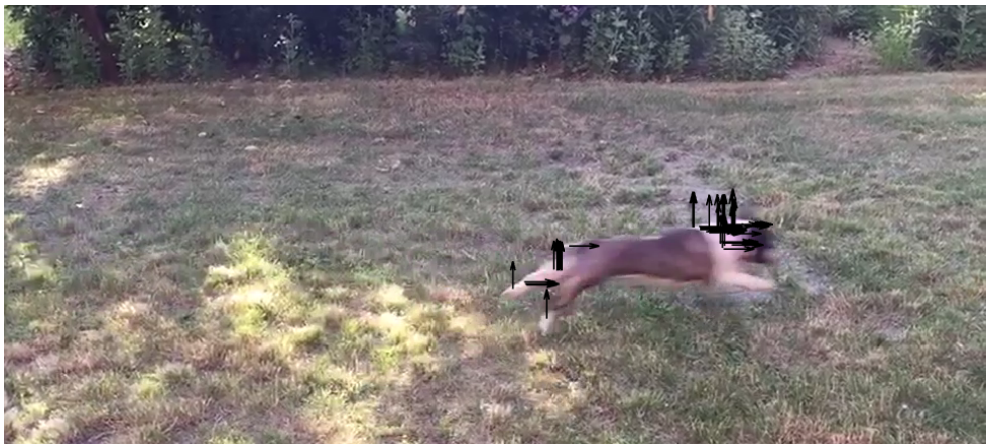


Figura 4.3: Flujo óptico obtenido por el método de Gauss-Seidel

4.2.2. Ejemplo 2

En este segundo ejemplo, el desplazamiento del coche determina el movimiento en la escena. Al igual que antes, el movimiento se produce en todas las direcciones espaciales, y se presenta principalmente en las partes fronterizas del coche.

En la figura 4.4 se encuentran las nuevas imágenes a analizar.

(a) Imagen inicial (b) Siguiente imagen en la secuencia

Figura 4.4: Intervalo de imágenes a analizar

El resultado de aplicar el método de Lucas-Kanade es el siguiente:

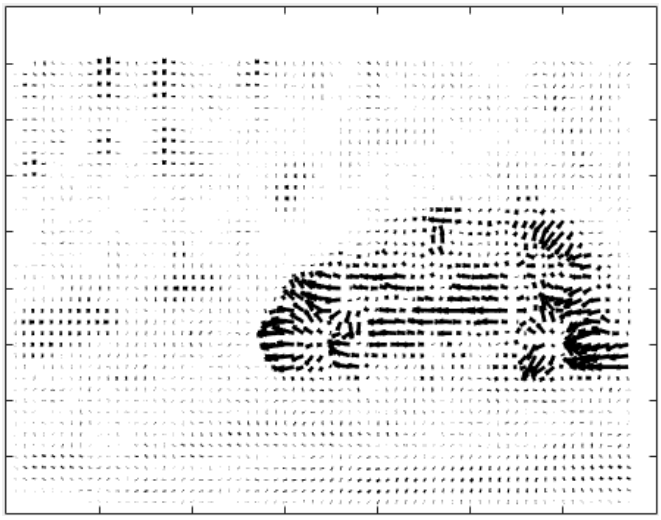


Figura 4.5: Flujo óptico obtenido por el método de Lucas-Kanade

Aplicando ahora el método global de Gauss-Seidel con parámetros $k = 3$, $\varepsilon = 10^{-}8$ y $\lambda = 1$, se obtiene el resultado del cómputo del flujo óptico mostrado en la figura 4.6. El movimiento se localiza principalmente en las partes significativas de la imagen, en relación al movimiento propiamente dicho (ruedas y partes trasera y delantera donde éste es relevante). Al igual que en el ejemplo anterior, las componentes de los vectores están orientadas hacia la dirección principal del movimiento.

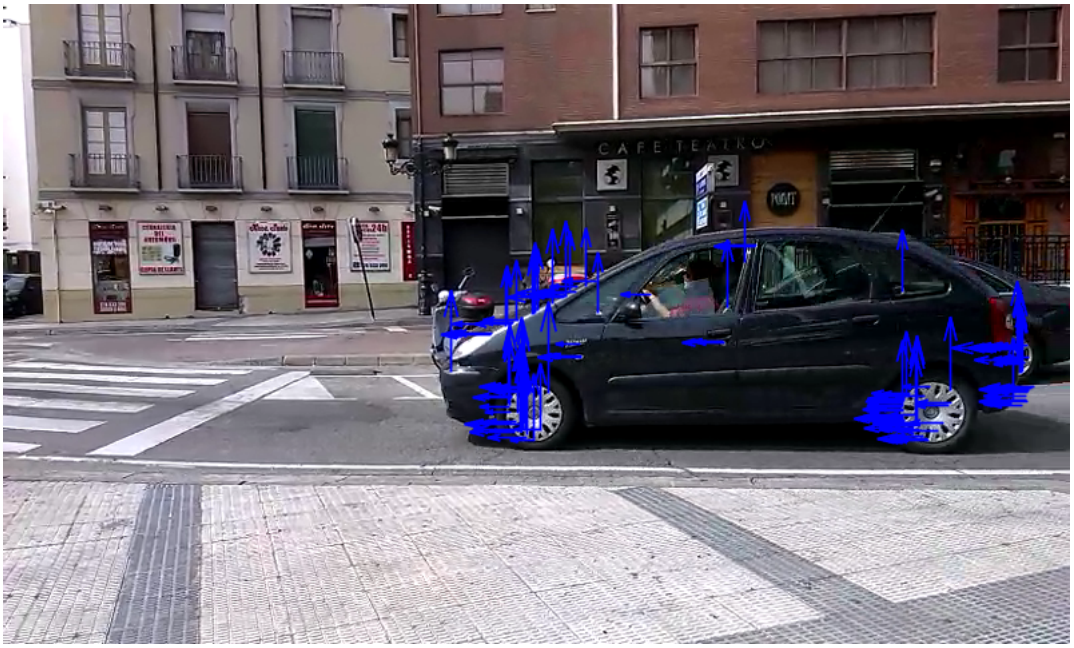


Figura 4.6: Flujo óptico obtenido por el método de Gauss-Seidel

Capítulo 5

Conclusiones y trabajo futuro

5.1. Conclusiones

La detección de movimiento en secuencias de imágenes con los métodos propuestos en este trabajo tiene numerosas aplicaciones en diversos ámbitos sociales e industriales. Sin embargo, cada método tiene sus restricciones en cuanto a continuidad, como ya se ha explicado en el capítulo 3. Esto ocurre también para los distintos procedimientos aplicados para el análisis del movimiento, algunos de los cuales se citan posteriormente en la sección 5.2.

El trabajo aquí presentado tiene un amplio ámbito de aplicación en diversos campos, pudiendo ser fácilmente adaptado para incorporarlo en sistemas más complejos. La opción de análisis automático proporcionada por la aplicación está de hecho enfocada a un análisis de vídeo en tiempo real.

Bajo estos planteamientos y premisas, en el presente trabajo se ha desarrollado una aplicación informática para detección del movimiento en secuencias de imágenes. Con tal propósito se han aplicado dos métodos específicos para la detección del movimiento, concretamente el método clásico de Lucas-Kanade y el de Gauss-Seidel. Ambos obtienen lo que se conoce como flujo óptico, si bien desde dos perspectivas. El primero de forma local y el segundo formulando un planteamiento más global, en este último caso tratando de resolver la problemática

surgida por la pérdida de la tercera dimensión en las imágenes, dada su naturaleza bidimensional, y que el primero no contempla para su solución.

Los métodos mencionados para el análisis del movimiento se han integrado en una aplicación informática, que se ha desarrollado bajo los planteamientos establecidos por la metodología propia de la Ingeniería del Software. En este sentido, se han definido los requisitos necesarios, se ha elaborado y seguido un plan de proyecto y se ha realizado un análisis de alto nivel, materializado en el correspondiente diseño e implementación, junto con las pruebas de verificación y validación. En todos los casos adaptados a las dimensiones del proyecto, pero que han permitido establecer perfectamente los planteamientos básicos para el desarrollo de una aplicación a nivel empresarial.

5.2. Trabajo futuro

El campo del análisis de movimiento abarca muchos aspectos, no sólo la detección de movimiento en la escena, sino también la localización de objetos en movimiento, con cámaras tanto estáticas como dinámicas, por ejemplo el estudio de imágenes por satélite para la detección de fenómenos naturales o meteorológicos, por citar algunos, o la predicción del tráfico, y la obtención de propiedades 3D a partir de secuencias de imágenes en dos dimensiones.

Dentro del estudio del movimiento, se pueden formular planteamientos de naturaleza geométrica, como es la formulación analítica del campo de movimiento, o diferenciales, como son las técnicas basadas en variaciones espacio-temporales. Por otro lado, otro ámbito es la detección de la profundidad a la que se encuentran los objetos y de colisiones, importante para la navegación de robots. Otra perspectiva es el análisis del movimiento a partir de puntos de interés, dando lugar a la aplicación de métodos de distinta naturaleza. Esto requiere un tratamiento y análisis previo de las imágenes, como puede ser encontrar puntos de borde o esquinas, con el posterior procedimiento de emparejar dichos puntos a lo largo de la secuencia.

Un campo importante, mencionado previamente, consiste en el seguimiento de objetos en la escena, de forma que mediante procedimientos específicos de predicción y medida es posible tal seguimiento. En esta línea se sitúa, por ejemplo, el conocido como filtro de Kalman, que incluye parámetros dinámicos con tal finalidad. Su estudio dentro del análisis del movimiento puede ser un aspecto a considerar en el futuro.

La presente aplicación podría ampliarse incorporando tratamiento previo de imágenes, mediante reducción de ruido, filtrado, extracción de bordes, etc. De esta forma mejorarían mucho los resultados obtenidos, particularmente con fines de su visualización. Otras posibles mejoras están descritas en el capítulo de requisitos en la sección 2.1.2 de requisitos futuros.

Finalmente, tal y como se ha mencionado previamente, esta aplicación se enmarca dentro del paradigma IoT, por lo que el desarrollo de la aplicación bajo los planteamientos del IoT constituye uno de los objetivos de futuro. Tal es el caso del desarrollo de la aplicación considerando aspectos tales como comunicación mediante Internet, transmisión de datos, procesamiento en local y remoto, así como planteamientos de "cloud computing" o incluso "Big Data", actualmente en continuo auge.

Apéndice A

Funciones

A continuación se exponen con detalle todas las funciones, correspondientes al módulo de Interfaz Gráfica.

Función SELECCIÓN DE DATOS DE ENTRADA.
Descripción elección del archivo de vídeo a analizar.
Entradas ruta completa del archivo (a través de un explorador).
Salidas mensaje de error si el archivo no es correcto (extensión incorrecta) y redirección a la pantalla de análisis.
Proceso el usuario buscará el archivo a través de un explorador de carpetas y lo seleccionará. El sistema lo abrirá y mostrará en la pantalla de análisis.
Origen interfaz.
Destino interfaz.
Precondiciones existencia del archivo y extensión compatible.
Postcondiciones incorporación al sistema del archivo sobre el que realizar el análisis.
Efectos laterales no existen.
Prioridad alta.

Función SELECCIÓN DE IMAGEN.
Descripción elección del fotograma inicial del vídeo para su análisis.
Entradas imagen del vídeo.
Salidas una imagen seleccionada.

Proceso se dejará seleccionada la imagen.
Origen interfaz.
Destino interfaz.
Precondiciones previa apertura del archivo de vídeo.
Postcondiciones dicha imagen será el punto de origen para el análisis.
Efectos laterales no existen.
Prioridad alta.

Función ANÁLISIS AUTOMÁTICO.
Descripción selección de todo el vídeo para su análisis.
Entradas vídeo.
Salidas vídeo seleccionado.
Proceso se guardará el vídeo completo como entrada.
Origen interfaz.
Destino interfaz.
Precondiciones previa apertura del archivo de vídeo.
Postcondiciones el vídeo completo será el objetivo del análisis. En caso de seleccionar Gauss-Seidel como método de análisis sólo será necesario introducir el parámetro λ.
Efectos laterales no existen.
Prioridad Prioridad: media.

Función SELECCIÓN DEL MÉTODO DE ANÁLISIS.
Descripción Elección del algoritmo para llevar a cabo el análisis.
Entradas método de Lucas-Kanade o Gauss-Seidel.
Salidas un método seleccionado.
Proceso el sistema guardará el nombre del algoritmo elegido.
Origen interfaz.
Destino interfaz.
Precondiciones previa apertura del archivo de vídeo.
Postcondiciones el algoritmo seleccionado será el usado durante el análisis.
Efectos laterales no existen.
Prioridad alta.

Función SELECCIÓN DE PARÁMETROS.
Descripción introducción de los parámetros necesarios para la ejecución del algoritmo de Gauss-Seidel.
Entradas número de iteraciones máximo, valor de ϵ y λ.
Salidas valores de los parámetros.
Proceso el sistema guardará los valores para el análisis.
Origen interfaz.
Destino interfaz.
Precondiciones selección del algoritmo de Gauss-Seidel.
Postcondiciones los valores introducidos serán los usados en el análisis.
Efectos laterales no existen.
Prioridad alta.

Función ANÁLISIS.
Descripción tiene lugar el análisis para la detección del movimiento.
Entradas datos seleccionados.
Salidas resultado del análisis.
Proceso el sistema aplicará el algoritmo seleccionado, con los parámetros indicados en caso necesario. Si se ha seleccionado análisis automático, se ignorará la imagen elegida. En caso de faltar algún dato se mostrará una ventana de error.
Origen interfaz.
Destino pantalla de resultados.
Precondiciones selección del algoritmo y parámetros en el caso de Gauss-Seidel, y si no se ha elegido análisis automático, una imagen del vídeo seleccionada.
Postcondiciones no existen.
Efectos laterales no existen.
Prioridad alta.

Apéndice B

Estimación del tamaño del proyecto

Siguiendo el modelo presentado en [8], los valores obtenidos para el presente proyecto son los siguientes:

Valor	VO	VPR	VPE	VE	Peso	Conteo PF
#Entradas	1	1	3	1.33	4	5
#Salidas	6	3	0	3	8	9
#Peticiones	6	2	1	3	7	8
#Archivos	6	2	1	3	5	8
#Interfaces externas	6	2	1	3	5	8
#Total						**85**

Tabla B.1: Estimación de información de valores de dominio.

VE se obtiene para cada elemento según la ecuación 2.1.

El total de los factores de ajuste de valor, F_i, es 25, obtenido tras responder a las preguntas propuestas en [8]. Las respuestas son: {1,0,0,1,4,5,1,2,1,2,3,3,0,2}. Sustituyendo en la ecuación 2.2 se tiene

$$PF = 85 * (0{,}65 + 0{,}01 * 25) = 77 \tag{B.1}$$

Así, el esfuerzo total en términos de persona-mes (pm), teniendo en cuenta una productividad organizacional promedio para este tipo de sistemas de 10 será, según la ecuación 2.3:

$$Esfuerzo = 77(PF)/10(PF/pm) = 8(pm) \tag{B.2}$$

Apéndice C

Manual de uso

En primer lugar, para abrir la aplicación se requiere la instalación de Matlab Runtime del sistema operativo correspondiente, proporcionado junto con la aplicación. Una vez instalado, basta ejecutar desde línea de comandos lo siguiente:

./**run_visio.sh** <**mcr_directory**>

donde <mcr_directory> es la ruta a la instalación de Matlab Runtime. En el caso de Linux, será similar a /usr/local/MATLAB/MATLAB_Runtime/v85, y en el caso de OS X, /Application/MATLAB/MATLAB_Compiler_Runtime/v84. Tras la ejecución de este comando, se ejecuta la aplicación.

Una vez abierta, se presenta la opción de abrir vídeo, como puede verse en la figura C.1. Al seleccionar esta opción se presenta al usuario una pantalla de exploración de carpetas (figura C.2), donde puede elegir el archivo de vídeo sobre el que realizar el análisis. En la parte inferior derecha se muestran las extensiones aceptadas por la aplicación en una lista desplegable.

Figura C.1: Pantalla de inicio

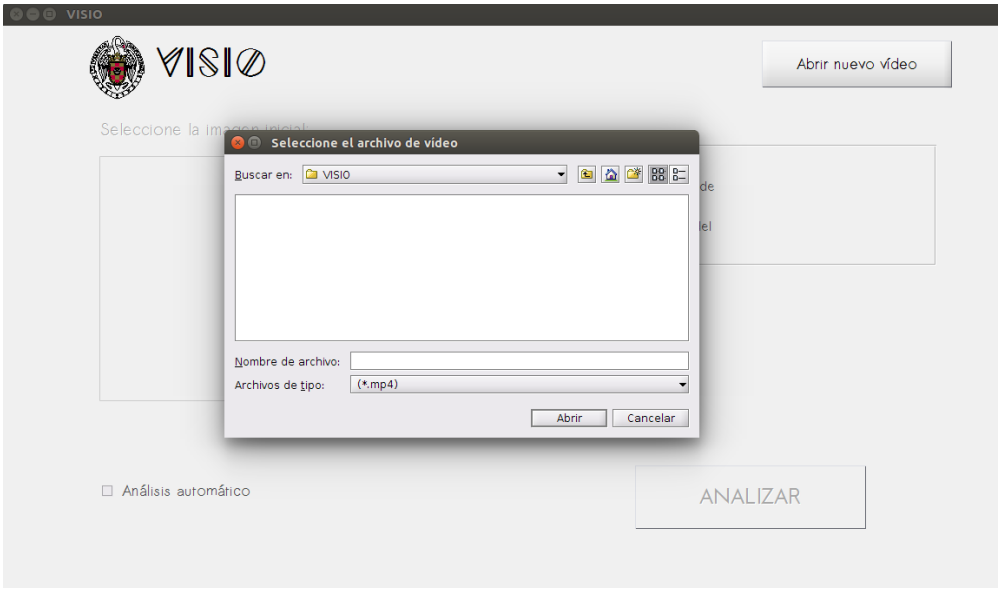


Figura C.2: Selección de vídeo

Una vez abierto el vídeo se habilitan las opciones disponibles, como puede verse en la figura C.3. Para seleccionar el fotograma inicial sobre el que realizar el análisis, arrastrar la barra de desplazamiento situada debajo de éste hasta la imagen deseada. En caso de activar la opción de análisis automático dicha imagen será descartada, pues se procesará el vídeo completo.

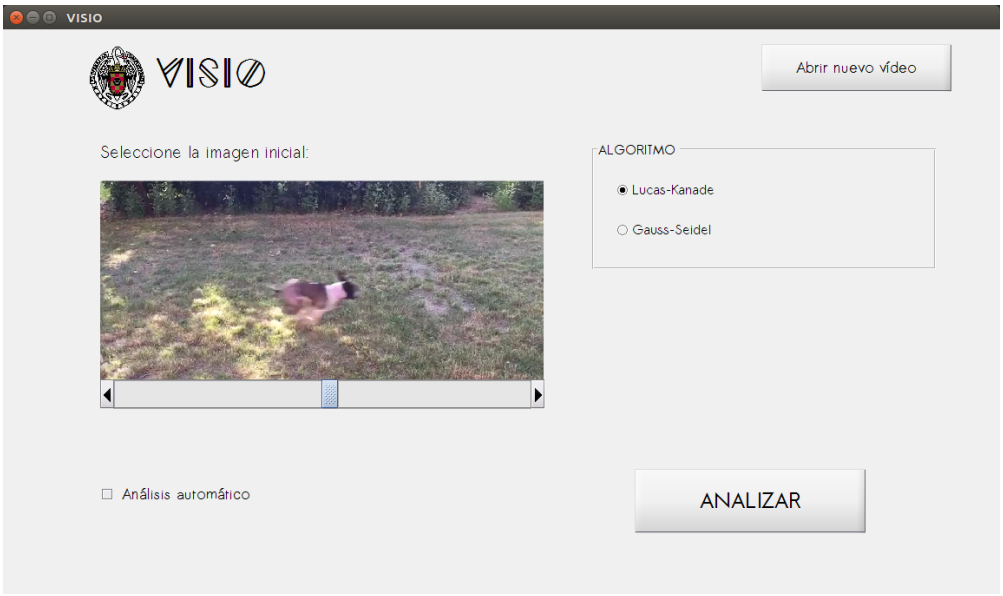
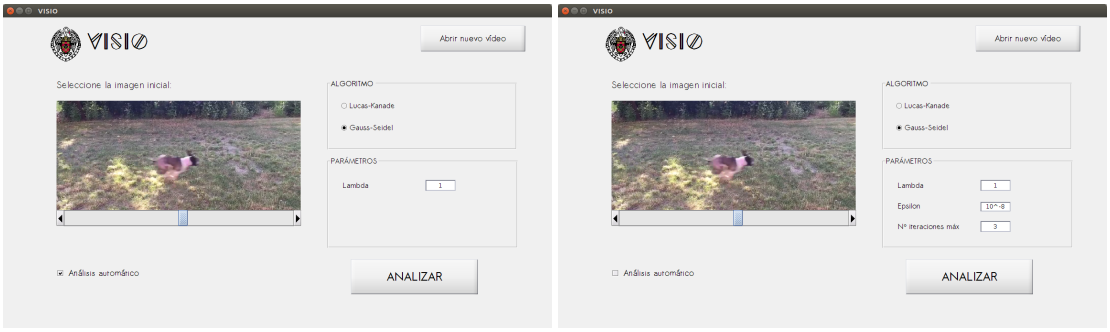


Figura C.3: Pantalla de análisis

En la parte derecha de la pantalla se encuentran los métodos disponibles para realizar el análisis: Lucas-Kanade y Gauss-Seidel. En el segundo caso se requerirá la introducción de parámetros: lambda (λ) en ambos modos de análisis (figura C.4a), y epsilon (ε) y el número de iteraciones únicamente si no se ha seleccionado la opción de análisis automático (figura C.4b). No se podrá realizar el análisis si estos parámetros no han sido introducidos. Los valores recomendados se muestran la siguiente imagen:



(a) Análisis automático (b) Análisis manual

Figura C.4: Introducción de parámetros para el análisis

Una vez seleccionado el método de análisis se podrá proceder al análisis con el botón destinado a ello situado en la parte inferior derecha de la pantalla. Los resultados se muestran en una nueva pantalla, como se puede ver en la figura C.5.

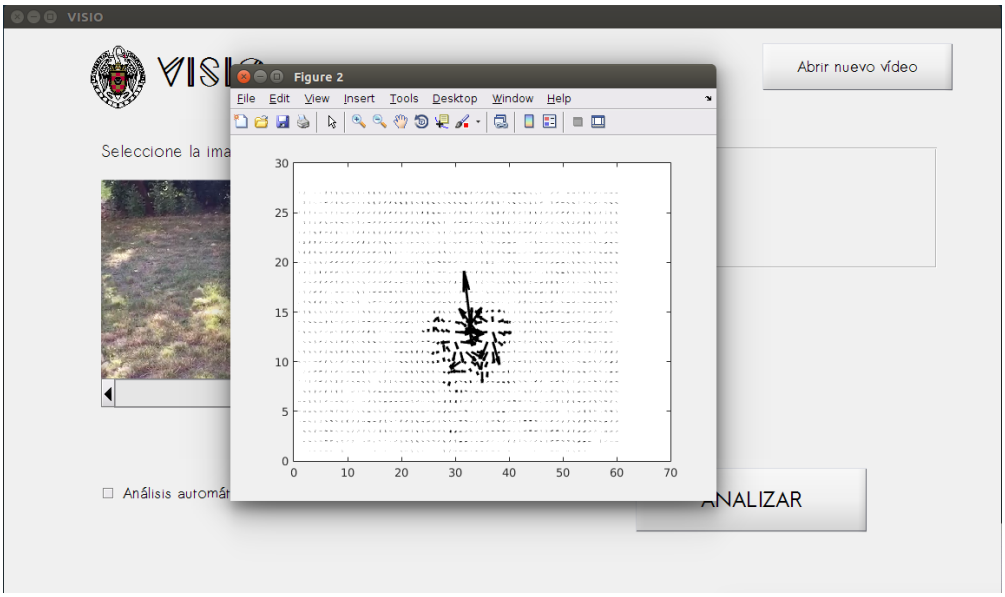


Figura C.5: Resultados del análisis

En cualquier momento se puede seleccionar un nuevo vídeo y comenzar de nuevo. Para cerrar la aplicación basta con usar el aspa de la parte superior de la pantalla.

Bibliografía

[1] The Matworks. http://www.mathworks.com/products/matlab/

[2] Matlab GUI. http://www.mathworks.com/discovery/matlab-gui.html

[3] Matlab Compiler. http://www.mathworks.com/products/compiler

[4] Matlab Runtime. http://www.mathworks.com/products/compiler/mcr/

[5] Software Engineering Standards Committee of the IEEE Computer Society, IEEE Recommended Practice for Software Requirements Specifications, 1998.

[6] Software Engineering Standards Committee of the IEEE Computer Society, IEEE Standard for Software Project Management Plans, 1998.

[7] G. Pajares, J. M. Cruz, Visión por computador: imágenes digitales y aplicaciones, RA-MA, 2007.

[8] Roger S.Pressman, Ingeniería del Software: Un enfoque práctico, McGraw Hill, 2010.

[9] Ian Sommerville, Ingeniería de Software, Pearson, 2011.

[10] G. Pajares, J.M. Cruz, Ejercicios resueltos de visión por computador, RA-MA, 2007.

[11] Microsoft Project Professional 2016. https://products.office.com/en-us/project/project-professional-desktop-software

[12] SQAS21.01.00-1999, Software Risk Management: A Practical Guide, Department of Energy Quality Managers Software Quality Assurance Subcommittee, 2000.

[13] Online Diagram and Flowchart Software. https://www.gliffy.com